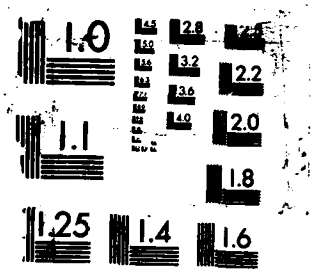
AD-A188 079 THREE-DIMENSIONAL MESH-GENERATION AND PLOTTING ROUTINES FOR USE WITH A BO..(U) ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) K W MAN ET AL. JUL 86 1/1

UNCLASSIFIED RAE-TM-MAT/STR-1077 DRIC-BR-100384 F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART

ROYAL AIRCRAFT ESTABLISHMENT

Technical Memorandum Mat Str 1077

Received for printing 8 July 1986

THREE-DIMENSIONAL MESH-GENERATION AND PLOTTING ROUTINES FOR USE WITH A BOUNDARY ELEMENT PROGRAM (BIE3D)

by

K. W. Man

M. H. Aliabadi*

SUMMARY

This Memorandum describes two computer programs which have been developed as data preparation tools for the three-dimensional boundary integral equation (BIE3D) computer program. The first is a three-dimensional mesh generation computer program which is based on a semi-automatic subdivision technique similar to a two-dimensional finite element mesh generation procedure. The program is written in standard FORTRAN and it is now operational in the Materials and Structures Departmental VAX computer. The second is a computer program for plotting the generated mesh. The plotting program is used to draw a three-dimensional view of the mesh of a problem. In particular, a generated mesh can be drawn so that its topology can be checked visually for anomalous points and incompatibility of elements. The program is written in BASIC and it runs interactively on a Hewlett-Packard HP2647F graphics terminal.

Both computer programs are described with examples. Instructions and descriptions for both input and output data file are presented in each case.

* Engineering Materials, University of Southampton.

LIST OF CONTENTS

1 INTRODUCTION

This Memorandum describes two computer programs which have been developed as data preparation tools for the three-dimensional boundary integral equation (BIE3D) computer program[1]. These programs will help BIE3D users to perform their tasks more efficiently and effectively. In particular, time saving is the biggest advantage of using these programs. The programs are called MESH3D and DRAW3D.

MESH3D is a three-dimensional mesh generation computer program for use with the three-dimensional boundary integral equation* computer program (BIE3D)[1]. It is written in standard FORTRAN and it is available in the Materials and Structures Departmental VAX computer.

DRAW3D is a computer program that can be used to plot a generated three-dimensional mesh on a graphics visual display or on paper. The program is written in BASIC and it can be operated interactively by the users on a Hewlett-Packard HP2647F graphics terminal.

These programs are now available to all BIE3D users as an option. The users can either create a BIE3D input data file in the conventional way, as described fully in a previous Memorandum[1], or use the MESH3D program as described here in section 2. A graphic output of the generated mesh can be obtained using the DRAW3D program, as described in section 3.

The main aim of this Memorandum is to lay down the guide lines of the operating procedures of the programs. Step-by-step instructions are given to show how an input and output data files can be prepared for both of the programs. Furthermore, test cases for both of the programs are illustrated with examples and descriptions.

2 MESH-GENERATION

In order to reduce the amount of time involved in data preparation for a boundary integral equation (BIE) analysis, computer-aided-design (CAD) software such as mesh generation computer programs can be employed. The use of such programs not only saves time but also eliminates the human errors which occur during manual preparation of the nodal co-ordinates and element topologies. Mesh generation programs have been well documented in the finite element method (FEM). These programs are generally of two types[2,3]:

(a) A digitizing tablet coupled with an interactive graphics system may be employed to define and produce the geometric data. A light pen is usually used to digitize the structural outlines of a preliminary mesh. The program may then allow for the choice of element types and the fineness of subdivision of elements. With this information, the mesh generation program will generate a mesh automatically and output it onto a graphics visual display or a plotter. However mesh generation algorithms of this type are highly dependent on the make and model of the digitizer and normally require specialised system software.

(b) A semi-automatic approach may be used, where the structural outline is based on a few large blocks which are defined as input data. Elements are then generated

---

* This method is also referred to as the Boundary Element Method (BEM).

automatically by a subdivision process. In this subdivision process, the generated element sizes need to be specified as a separate set of input data. This approach is described in detail in the following sections and it is the method used in the present study. A two-dimensional finite element mesh generation routine[4] which is based on this approach has been adopted and modified for generating a mesh on a two-dimensional surface, as required for a three-dimensional BIE calculation.

2.1 The mesh-generation computer program (MESH3D)

This three-dimensional mesh-generation program has been developed jointly with the Department of Engineering Materials, University of Southampton and is now installed in the Materials and Structures Departmental VAX computer. The program (MESH3D) is based on automatic element subdivision of a few large structural blocks which are defined as input data: each block is subdivided into elements according to weighting factors provided as input data.

MESH3D will calculate all the nodal co-ordinates for all the subdivided elements. The nodal point numbering convention (clockwise or anticlockwise), generated for the mesh, will be the same as the input data numbering convention for the structural blocks.

2.2 Geometric representation of a structural block

In the method of BIE, the boundary contour is the surface of a three-dimensional body and is always represented by surface elements termed boundary elements. The type of boundary element used in BIE3D is the eight-node isoparametric quadrilateral element, as shown in Fig 1. MESH3D uses the eight-node quadrilateral to represent a structural block in the initial stage of the mesh generation process. The mathematical representation of a structural block is based on the global cartesian co-ordinate system where each of the nodes $\underline{x}^{\alpha}$ is approximated by interpolation formula which involves finite element shape functions as shown below:

$$\underline{x}(\xi,\eta) = \sum_{\alpha=1}^{8} N^{\alpha}(\xi,\eta)\underline{x}^{\alpha} \tag{1}$$

where $\underline{x}^{\alpha}$ are the nodal co-ordinates and $N^{\alpha}(\xi,\eta)$ are the shape functions[4] for a two-dimensional finite element analysis $\underline{x} = (x,y,z)$.

This interpolation is equivalent to the transformation of a boundary element to a plane square in a local co-ordinate system (ξ,η), as shown in Fig 1. For example, in Fig 1, the line $\xi = +1$ will map into the global co-ordinate system such that nodes 5, 4, 3, will lie on the same curve: this is an approximation to the actual shape of the edge of the boundary element. Other lines ξ,η = constant will similarly map into curves in the global co-ordinate system to approximate the boundary element, which is being parametrically represented by ξ,η .

2.3 The subdivision process

Any one eight-node quadrilateral element may be subdivided into a number of smaller elements and the sizes of the subdivided elements are predetermined by weighting factors specified as input data.

When an element is subdivided extra nodes will be created in order to restore the compatibility of the eight-node elements. For example, in subdividing a three-dimensional body, such as a cube in Fig 2, the subdivision of one of the elements on one side of the cube causes incompatibility of nodal points with the elements adjacent to it. Further additional nodal points, in all a total of twelve, must be introduced in order to restore the compatibility of the elements. In fact, the subdivision process not only subdivides a large block into smaller elements but also ensures all the elements and all the nodal points are compatible with each other; repeated common nodes at boundary interfaces are eliminated (see section 2.4). In order to achieve this, the subdivision process requires a set of input data that itself is consistent and compatible. For instance, for each structural block it is necessary to specify the following:

(a) The number of subdivisions to be made in both ξ and η directions are stored in NDIVX and NDIVY respectively. For example, if NDIVX = 3 and NDIVY = 2 then the block will be subdivided into three divisions in the ξ direction and two divisions in the η direction respectively, *ie* a total of six elements will be generated as shown in Fig 3. The direction of the subdivisions must be clearly defined where there are two or more blocks linked together because the compatibility at the block-to-block interfaces must be preserved, *ie* the number of elements at the adjacent blocks must be the same, as indicated in Fig 4.

(b) Weighting factors are specified as input data, for each individual block, to define the relative sizes of the elements within a block. The weighting factors are denoted by $(W_\xi)_i$ where ξ is the direction of the weighting factors. This information is stored in the following format:

(WEITX(IDIVX,(IDIVX=1,NDIVX) .

Similarly, the weighting factors $(W_\eta)_i$ in the η direction are stored as:

(WEITY(IDIVY),(IDIVY=1,NDIVY) .

It should be noted that these weighting factors are used to define the relative sizes of the elements in a particular way for a particular block; therefore their absolute values do not affect the mesh generation process. For example, in order to subdivide a large structural block into three elements the weighting factors $(W)_i$ may be specified as:

W_1	W_2	W_3
10	10	20

or

1	1	2

or

0.2	0.2	0.4 .

However, it is equally important to specify all the weighting factors consistently for all blocks adjacent to each other where compatibility of nodes at block-to-block interface must be reserved. A pictorial illustration of compatibility of nodes at block-to-block interfaces in shown in Fig 4.

TM MS 1077

Both local co-ordinates ξ and η are initialized to the value of -1, and are then incremented according to

$$\xi_i = \xi_{i-1} + \frac{(W_\xi)_i}{W_\xi^T} , \qquad (2)$$

$$\eta_i = \eta_{i-1} + \frac{(W_\eta)_i}{W_\eta^T} ,$$

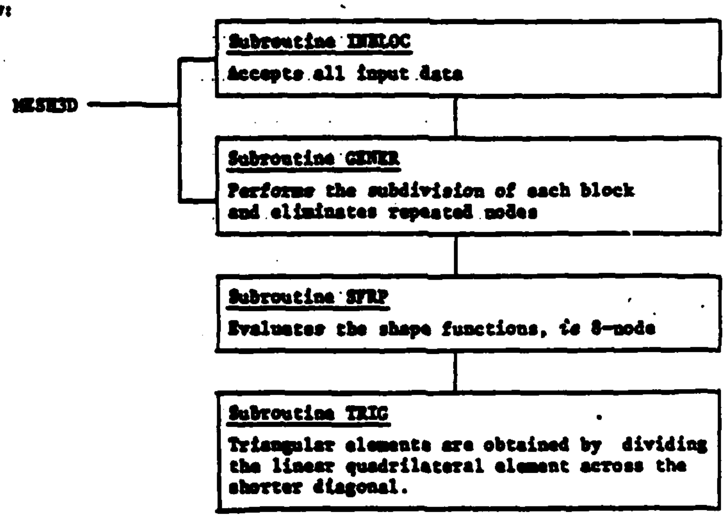
where $$W_\xi^T = \sum_{i=1}^{i=NDIVX} (W_\xi)_i \quad \text{and} \quad W_\eta^T = \sum_{i=1}^{i=NDIVY} (W_\eta)_i \qquad (3)$$

and where subscript i denotes the general increment and the summation is made over the total number of subdivided elements in the specified direction. The real space co-ordinates are then calculated using (1).

2.4 Connectivity of individual blocks

The subdivision process, as described in the previous section, is performed individually for each structural block. The generated element nodal points within each block are numbered individually. However element nodal points between block-to-block interfaces will have common co-ordinate values so they need to be uniquely numbered. This is done automatically by searching for all the points having common co-ordinate values and replacing the nodal numbers by a single nodal number. It should be noted that this process of elimination of repeated nodes will in general produce a new nodal numbering system for the generated mesh, even for the case where all the structural blocks remain undivided; an example of this is illustrated in Fig 5a&b.

A typical mesh-generation routine may be illustrated using the simple flow diagram below:



2.5 Input data

Subroutine INBLOC reads in the data necessary to define a preliminary mesh which may contain several structural blocks. In the MESH3D input data file, apart from the problem title, all data may be entered as free-format.

Descriptions of a MESH3D input data file is given below and in order to see how this works a test case is illustrated in section 2.6 with the entire data file of the problem listed with descriptions. The users may use this example as a guide line to prepare a data file for the MESH3D computer program.

For clarity, the MESH3D input data file can be arbitrarily divided into seven groups as described below. It should be noted that a maximum of 750 elements may be generated as a mesh without altering the dimensions of the arrays in the program.

Group (1) PROBLEM TITLE:

* Enter problem title occupying not more than 80 characters.

Group (2) GAUSSIAN QUADRATURE

* Enter the order of Gaussian quadrature formula used for the integration over each element and the corresponding Gaussian abscissae and coefficients. (Note: The Gaussian abscissae and its coefficients are used in the Gaussian quadrature formulae to evaluate various integrals over elements and cells in BIE. A short summary of their values is listed in the Appendix or a comprehensive set of the coefficients may be found in Ref 5. As a guide line fourth- or sixth-order of Gaussian Quadrature is sufficient for most cases. If in doubt, the safest way to get information on the stability of the results is to make several runs, at the subsequent stage of using BIE3D, with different order of Gaussian quadrature. It should be noted that the Gaussian quadrature is not used in the algorithm of the mesh-generation program so they can be neglected here by entering a ZERO instead; however it will need to be changed for the BIE3D run.)

Group (3) MATERIAL PROPERTY

* Enter the Young's modulus and Poisson's ratio. (Note: Neither Young's modulus nor Poisson's ratio are used in the algorithm of the mesh-generation program; their values will not therefore affect the mesh result but are needed for BIE3D.)

Group (4) MESH DATA

* Enter the total number of points, total number of blocks, type of elements to be generated (*eg* eight) and the dimension of the co-ordinate system used (*eg* two or three). These data are read in subroutine INBLOC and the parameters defined in the program as:

  NPOIN - Number of co-ordinate points defining the structural outline.
  NELEM - Number of blocks (maximum of 50 blocks may be specified).

NNODE - Types of element which are permitted in the program are:

(a) Quadratic eight node isoparametric elements,

(b) Linear four node elements,

(c) Linear three-node triangular element.

NDIME - Number of co-ordinate dimensions (*eg* two or three).

Group (5) Structural block data

* Enter the block number, the numbers of the nodal points of the block. (Note the numbering convention for a block is as follows: as the surface is viewed along the inward normal from outside the body, the nodal numbers are specified in clockwise order, *ie* right-hand-screw rule.)

These parameters are defined in the program as:

NUMEL - Structural block number.

LNODS - Structural block data. (LNODS(NUMEL,INODE), INODE=1,NNODE), *ie* the block topology definition.

Group (6) Co-ordinates of the nodal points

* Enter the point number and the three-dimensional co-ordinate (x,y,z) of the nodal points of the blocks. These parameters are defined in the program as:

JPOIN - point number.

COORD(JPOIN,IDIME)(IDIME=1,NDIME) - (x,y,z) co-ordinates of the points.

(Note: maximum of 200 points may be used to define the structural outline without altering the size of the array dimension in the program.)

Group (7) WEIGHTING FACTORS

* Enter block number, number of elements in the ξ direction and number of elements in the η direction.

* Enter the corresponding weighting factors for the ξ direction and the weighting factors for the η direction.

These are the input data necessary to subdivide each structural block into smaller elements and these parameters are defined in the program as:

KBLOC - Block number.

NDIVX - Number of elements in ξ direction into which the block is to be subdivided.

NDIVY - as NDIVX but in ξ direction.

(Note: maximum value of four may be specified for NDIVX and NDIVY without altering the array dimension in the program.)

END-OF-FILE

2.6 Test case

A flat plate with a normal central hole is used here as an example to demonstrate the operating procedures of the mesh generation program MESH3D. The dimensions of the flat plate are shown in Fig 6. In order to make the problem as simple as possible the symmetry about the x- and y-axes is used so that only a quarter of the plate is modelled

in this example. The initial mesh of the flat plate contains a total of 10 large structural blocks and a total of 32 points as shown in Fig 7 and Fig 8.

In order to generate a mesh the large structural blocks are subdivided using the weighting factors as described in section 2.3. A listing of the input data file for the MESH3D program is listed below with descriptions. The data file name is designated as FP.DAT:-

Data file FP.DAT	Descriptions
TITLE: FLAT PLATE WITH A CENTRE HOLE	.... Problem title
6 .23861919 .46791393 .66120940 .36076158 .93246949 .17132449 -.23861919 .46791393 -.66120940 .36076158 -.93246949 .17132449	.... order of Gaussian quadrature and coefficients.
1000.0 0.3	.... Young's modulus and Poisson's ratio
32 10 8 3	.... Total number of points, total number of blocks, element type (ie 8) and co-ordinate dimensions.
1 1 2 3 4 5 6 7 8 2 5 4 3 9 10 11 12 13 3 12 11 10 14 15 16 17 18 4 17 16 15 19 20 21 22 23 5 7 6 5 13 12 24 25 26 6 25 24 12 18 17 23 22 27 7 7 26 25 28 29 30 1 8 8 25 27 22 21 20 31 29 28 9 20 19 15 14 10 32 29 31 10 1 30 29 32 10 9 3 2	.... Block number and block data specified in the right-hand-screw rule
1 1 0 0 2 1 0 .1 3 1 0 .2 4 1 .4 .2 5 1 .8 .2 6 1 .8 .1 7 1 .8 0 8 1 .4 0 9 .5 0 .2 10 0 0 .2 11 .4293 .4293 .2 12 .8586 .8586 .2 13 .9235 .8152 .2 14 0 .5 .2 15 0 1 .2 16 .4 1 .2 17 .8 1 .2 18 .8152 .9235 .2 19 0 1 .1 20 0 1 0 21 .4 1 0 22 .8 1 0 23 .8 1 .1 24 .8586 .8586 .1	.... Point number and the three-dimensional cartesian co-ordinate system

```
25  .8586 .8586 0
26  .9235 .8152 0
27  .8152 .9235 0
28  .4293 .4293 0
29  0 0 0
31  0 .5 0
32  0 0 .1

1  3  4                    .... Block (1) and weighting
3.5 3 3.5                  factors in the  ξ and η direction.
2 3.5 3 1.5

2  4  1

1.5  3  3.5  2             .... Block (2) and weighting
1                          factors in the  ξ  and η direction.

3  4  1                    .... etc
1.5  3  3.5  2
1

4  4  3

1.5  3  3.5  2
3.5  3  3.5

5  3  1
3.5  3  3.5
1

6  3  1
3.5  3  3.5
1

7  1  4
1
1.5  3  3.5  2

8  1  4
1
1.5  3  3.5  2

9  3  1
3.5  3  3.5
1

10  1  3
1
3.5  3  3.5
END                        .... end-of-file
```

This data file (FP.DAT) may then be submitted to the MESH3D program using the commands as follows:

```
$assign FP.DAT   for003     (assign to logical unit 3)
$assign FP.MES   for004     (assign to logical unit 4)
$run MESH3D
```

The input-data (FP.DAT) will be checked for the total number of nodal points and elements to ensure the array dimensional sizes in the MESH3D program will not be exceeded. An excessive value of these numbers will cause an execution error and an error message will be displayed before aborting the program. If the total number of nodal points and elements is not too large then the mesh coordinates of the three-dimensional mesh will be automatically generated according to the process of sub-division and the corresponding weighting factors. The output data file containing the generated mesh data of the quarter flat plate is called (FP.MES). Before this file can be submitted for the BIE3D run there are two more steps the user may need to carry out:

(1) visual inspection of the topology of the generated mesh, and

(2) specify boundary conditions.

The visual inspection of the mesh can be done once the mesh is plotted on paper. The procedures for the plotting of the mesh on paper are described in section 3. Once the mesh has been checked for misalignment, anomalous points or any incompatibility of element distributions etc it can then be modified to include the boundary conditions of the problem; the boundary conditions are inserted according to the format specified for boundary conditions in the BIE3D users' guide[1]. Finally the user can submit the modified (FP.MES) input data file, (MODFP.MES) say, and run the BIE3D program using the commands as follows:

```
$assigne MODFP.MES      for005      (assign to logical unit 5)
$assign FP.RES          for006      (assign to logical unit 6)
$RUN BIE3D
```

where 'FP.RES' is the output-data-file of the BIE3D program containing the displacements, the tractions and the surface stresses for all the nodes of the three-dimensional mesh of the quarter-plate.

3 MESH PLOTTING

The checking of the nodal co-ordinates of three-dimensional mesh data is often tedious and difficult. In particular, if the mesh data has been generated by a mesh generation program, the large volume and the three-dimensional complexity of the mesh data would make the checking process an awesome task. The obvious alternative approach is to plot the generated mesh on paper in an isometric view. The visual topology of the mesh would allow for the checking of misalignment, anomalous points or any incompatibility of element distributions etc. Such a visual presentation for a mesh in three dimensions is invaluable and absolutely essential for any mesh design and mesh generation work. Indeed, during the plotting process, if one follows the path of the plotting pen on the chart, one can easily determine without recourse to the numerical data file, whether the mesh has been generated in the specified numbering convention. Furthermore, with the option of having all the nodal numbers of all the element numbers or both plotted, the numbered mesh can be used to assist the specification of all the boundary conditions in which each individual nodal point or surface element can be precisely located and constrained if required.

A three-dimensional mesh plotting computer program (DRAW3D) has been written and implemented in the local computer system. The program is written in BASIC and it runs interactively on a Hewlett-Packard HP2647F graphics terminal. It can either be used to plot a three-dimensional mesh, in an isometric view, on a graphic terminal display or on paper. In this section, the handling of input and output data files will be explained and how exactly the graphic plotting programs can be used in conjunction with the mesh generation program (MGEN3D) and the boundary integral equation program (BIE3D).

3.1 The graphic plotting computer program (DRAW3D)

The computer program DRAW3D can be used, on a Hewlett Packard HP2647F terminal, to plot a three-dimensional view of a mesh. The plotting program (DRAW3D) is now available in the local computer system. It has been extensively used throughout the present work and shown to be useful for checking of a generated mesh. The program can be used interactively by the users so that the view angles of an isometric plot can be rotated through 0 to 360 degrees in any axes to show any side of the figure. Also, the dimensions of the figure in the x- and y-planes can be distorted so that exaggeration of any particular plane of the figure may be made. When the final projection of the configuration is accepted, it can be plotted on paper as a permanent record.

3.2 Data preparation for the plotting program

A BIE3D input data file requires very little change before it can be submitted for a plotting program run. In fact the only change which is required is to convert a FORTRAN input data file format into a BASIC input data file format. The conversion is an essential step and it is done by separating each data item by a comma.

The following section will list all the steps required to prepare a plotting program input data file.

3.3 Operating procedures for the graphic plotting program

(1) Separate the mesh data in a BIE3D input data file by commas. This can be achieved either by typing the commas in by hand or submitting the input data file to a utilities program called READCON in the VAX computer. The utilities program will separate all the data in the data file with commas automatically:

```
$assign (BIE3D input data file )     FOR001
$run READCON
```

and the output of the modified mesh data is assigned to the display.

(2) Now we can transfer this modified mesh data from the display to a floppy disc,

(3) and then run the BASIC plotting program by typing in
> GET "DRAW3D" (assuming that the plotting program is already in the disc)
> RUN

At this point, the plotting program will respond with prompts in an interactive mode and a series of prompts will be displayed on the screen:

> Plotting device? (display=0 or plotter=5)
> Input file name:
> Element type? (Triangular=3 or Quadrilateral=8)

Once these questions are answered, the program will take over and plot out the three-dimensional view of your mesh. After it has finished, further prompts will appear on the display:

> Do you wish to have the element-number labelled? (Y or N)
> Do you wish to have the nodal-number labelled? (Y or N)

When these are done the mesh plotting process is completed and the program will re-start with the following prompts:

> ENTER view-angles (rotation about x- and y-axes) and distortion in y direction:
> ENTER Plotting device? (display=0 or plotter=5)
> Input total number of hidden elements:
> Do you want to plot distorted shape? (Y or N)

The user can select any one of these options and carry on as before.

3.4 Test case

The file called (FP.MES) is the generated mesh data file for the quarter-plate example as described in section 2.6. It is used here to demonstrate the operating procedures of the plotting computer program DRAW3D.

Before the data file can be submitted for the plotting run, the format of the data file must be converted to the BASIC data format as described in section 3.3 The converted data of file (FP.MES) is listed below: it should be noted that this listing has been condensed to save space.

Two three-dimensional presentations of the generated mesh are plotted on paper as shown in Fig 9 and Fig 10. It can be seen that a total of 52 quadrilateral elements and a total of 158 nodal points have been generated in this case. The element numbering system and nodal numbering system of the generated mesh are shown in Figs 9 and 10, respectively.

Data file	Descriptions
TITLE: FLAT PLATE WITH A CENTRE HOLE	.... Title
158, 52	.... Number of nodes and number of elements.
1, 1.000000 , 0.0000000+00, 0.0000000E+00	.... Node number and
2, 1.000000 , 0.0000000E+00, 3.5000000E-02	x,y,z cartesian co-ordinates
3, 1.000000 , 0.0000000E+00, 7.0000000E-02	
. . . .	
. . . .	
. . . .	
155, 0.000000 , 0.0000000E+00, 0.1300000	
156, 0.000000 , 0.0000000E+00, 0.1650000	

```
157,  0.500000  ,   0.0000000E+00 ,   7.0000000E-02
158,  0.500000  ,   0.0000000E+00 ,   0.1300000
 1,   1,   2,   3,   9,   14,  13,   12,   8           .... Element number and element
 2,   3,   4,   5,   10,  16,  15,  14,   9            data specified in right-hand-
 3,   5,   6,   7,   11,  18,  17,   16,   10          screw convention.
 4,  12,  13,  14,  20,  25,  24,  23,  19
 .    .    .    .    .    .    .    .    .
 .    .    .    .    .    .    .    .    .
 .    .    .    .    .    .    .    .    .
 .    .    .    .    .    .    .    .    .
48, 107, 98, 93, 151, 155, 154, 153, 150
49,  93, 84, 79, 70, 65, 156, 155, 151
50,   1, 144, 145, 152, 153, 157,  3,  2
51,   3, 157, 153, 154, 155, 158,  5,  4
52,   5, 158, 155, 156,  65,  56,  7,  6
35,  20,  1                                          .... view angles and distortion.
END                                                  .... End of file.
```

4 CONCLUDING REMARKS

The instructions contained in this Memorandum will enable users to run the programs MESH3D and DRAW3D as they stand, to obtain respectively a mesh and a graphics presentation of a mesh of a simple or complex three-dimensional configuration.

A three-dimensional mesh presentation of a real structure can be very complex and difficult to handle manually, but by using these programs the users can leave the number manipulation to the programs and concentrate on the accuracy of the modelling of the problem.

Appendix

Gaussian abscissae	Gaussian coefficients	
4	0.3399810435	0.6521451548
	0.8611363115	0.3478548451
	−0.3399810435	0.6521451548
	−0.8611363115	0.3478548451
6	0.2386191860	0.4679139345
	0.6612093864	0.3607615730
	0.9324695142	0.1713244923
	−0.2386191860	0.4679139345
	−0.6612093864	0.3607615730
	−0.9324695142	0.1713244923
8	0.1834346424	0.3626837833
	0.5255324099	0.3137066458
	0.7966664774	0.2223810344
	0.9602898564	0.1012285362
	−0.1834346424	0.3626837833
	−0.5255324099	0.3137066458
	−0.7966664774	0.2223810344
	−0.9602898564	0.1012285362

REFERENCES

No.	Author	Title, etc
1	K.W. Man	A user's guide for BIE3D - a boundary integral equation computer program for three-dimensional elastostatic analysis. RAE Technical Memorandum Mat/Str 1078 (1986)
2	O.C. Zienkiewicz D.V. Phillips	An automatic mesh generation scheme for plane and curved surfaces by isoparametric coordinates. *Int. J. Num. Meth. Engng.*, 3, 519-528 (1971)
3	W.R. Buell B.A. Bush	Mesh generation - A survey journal of engineering for industry. Trans. American Society of Mechanical Engineers series B, Vol 95, No.1
4	E. Hinton D.R.J. Owen	Finite element programming. Academic pre~s, London (1977)
5	P.C. Hammer A.H. Stroud	Numerical integration over simplexes and cones. *Math. Tables Aids Comp.*, 10, 130-137 (1956)

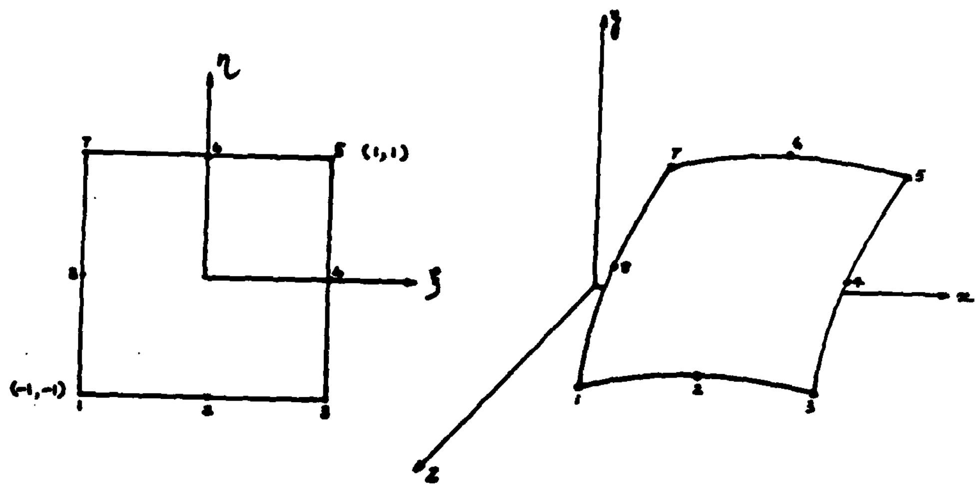


Fig 1 Eight-node quadrilateral element

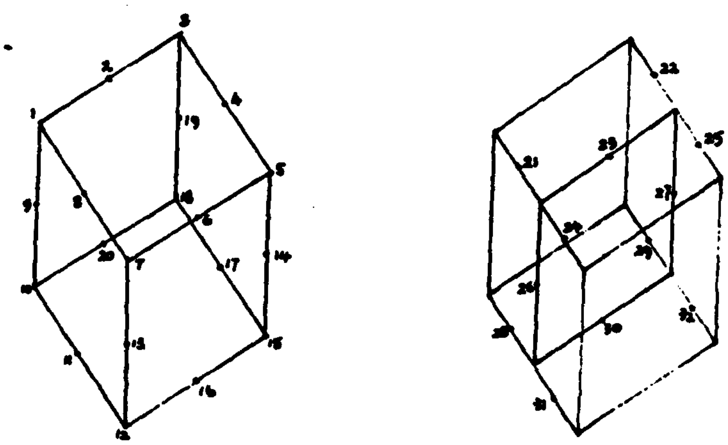


Fig 2 Subdivision of a three-dimensional body

Fig 3

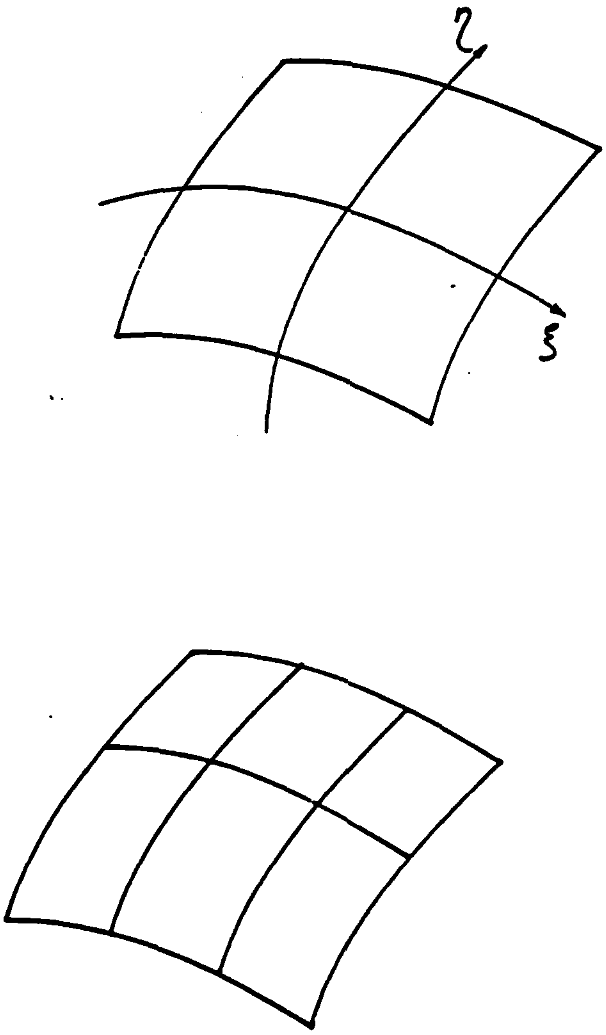


Fig 3 Subdivision of a structural block in ξ and η direction

TM MS 1077

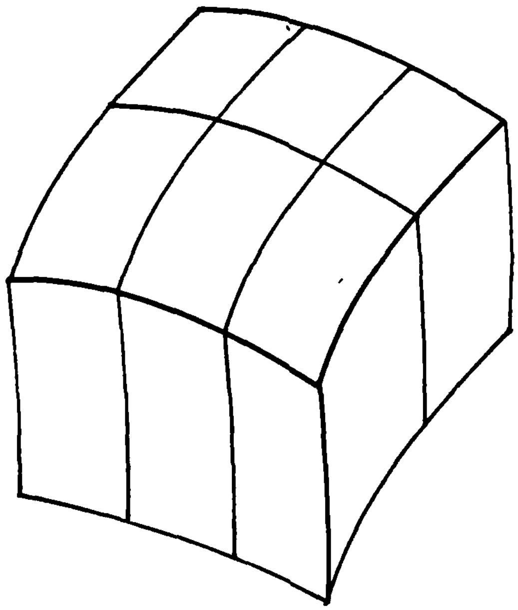
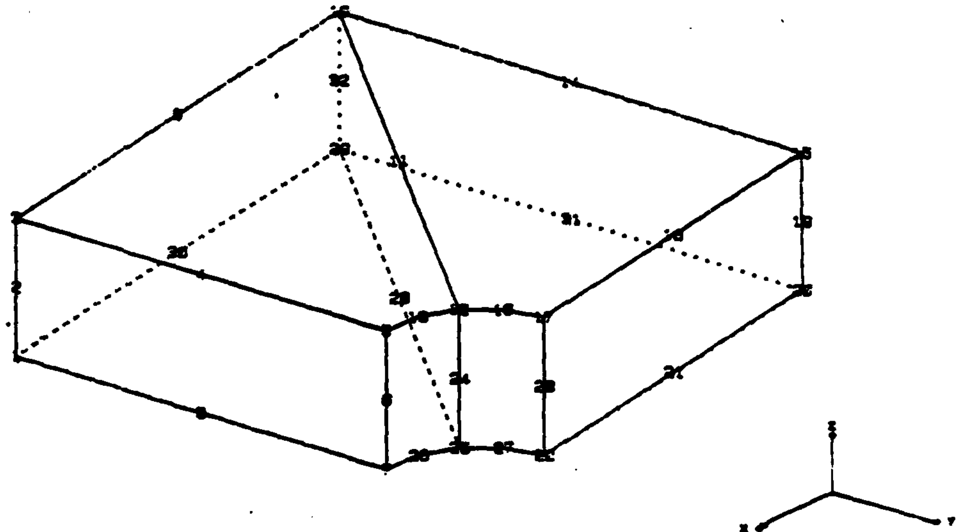
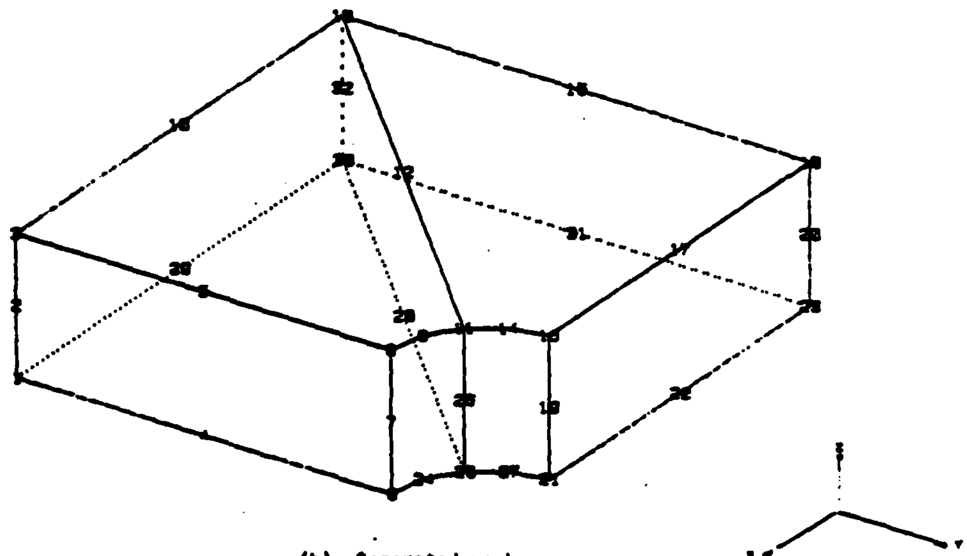


Fig 4 A pictorial representation of element and nodal compatibility at the block-to-block interfaces

Fig 5a&b



(a) Preliminary mesh



(b) Generated mesh

Fig 5 Nodal numbering system of the preliminary mesh and the undivided generated mesh

TM MS 1077

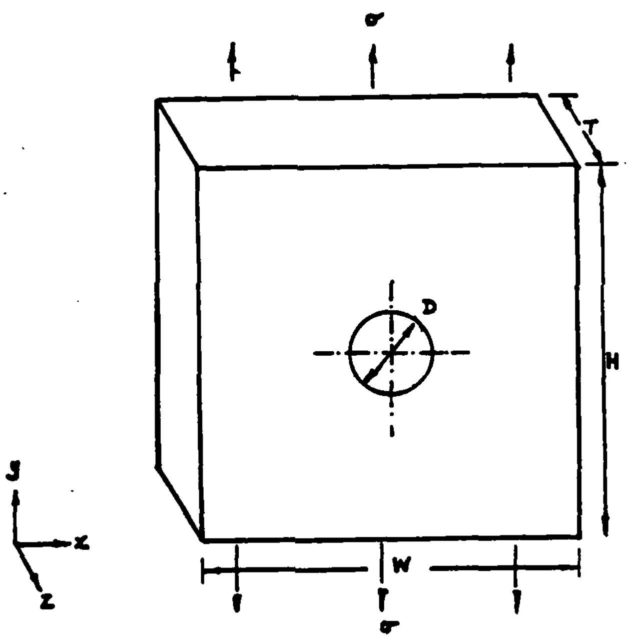


Fig 6 A flat plate with a central normal hole

Fig 7

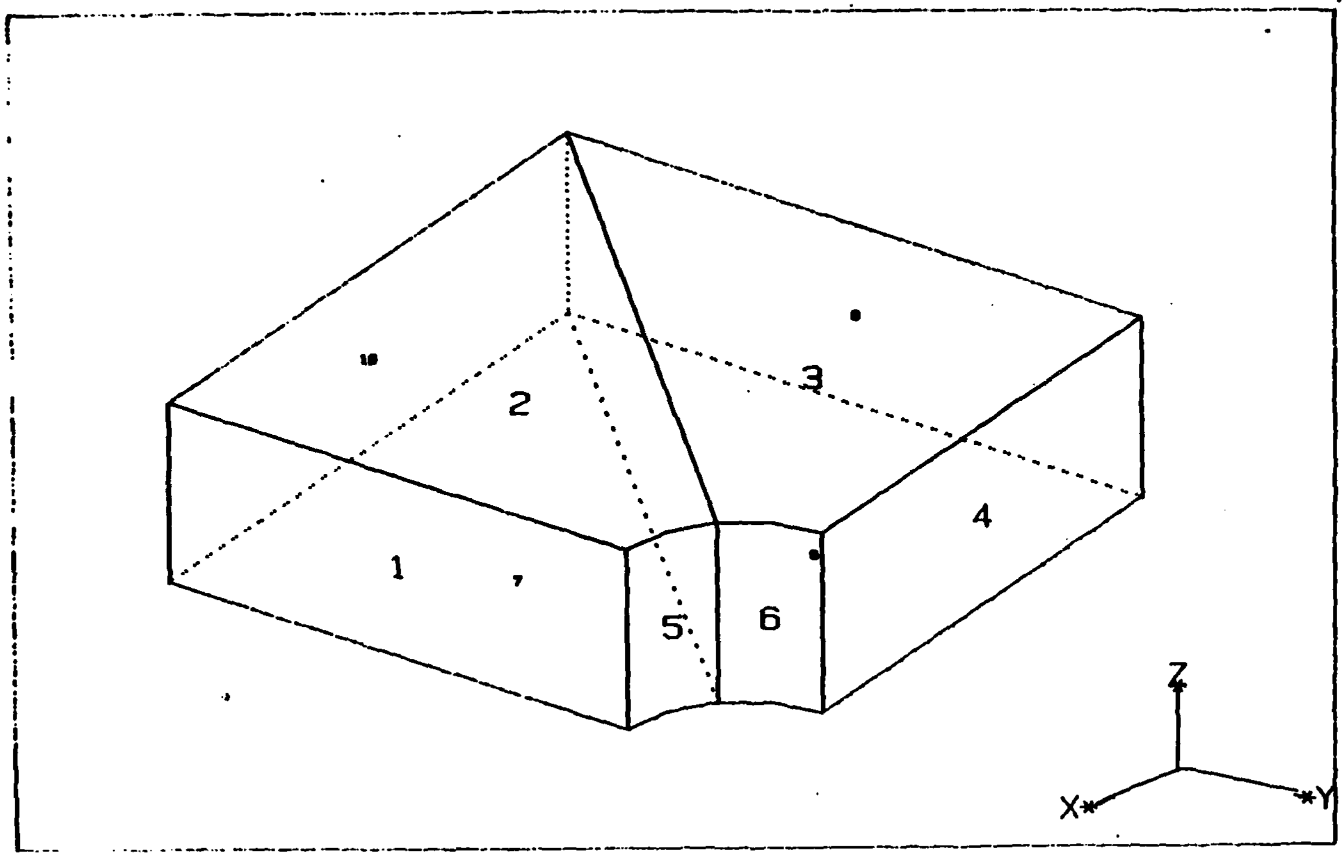


Fig 7 Preliminary mesh design and element numbering system of a quarter of the flat plate

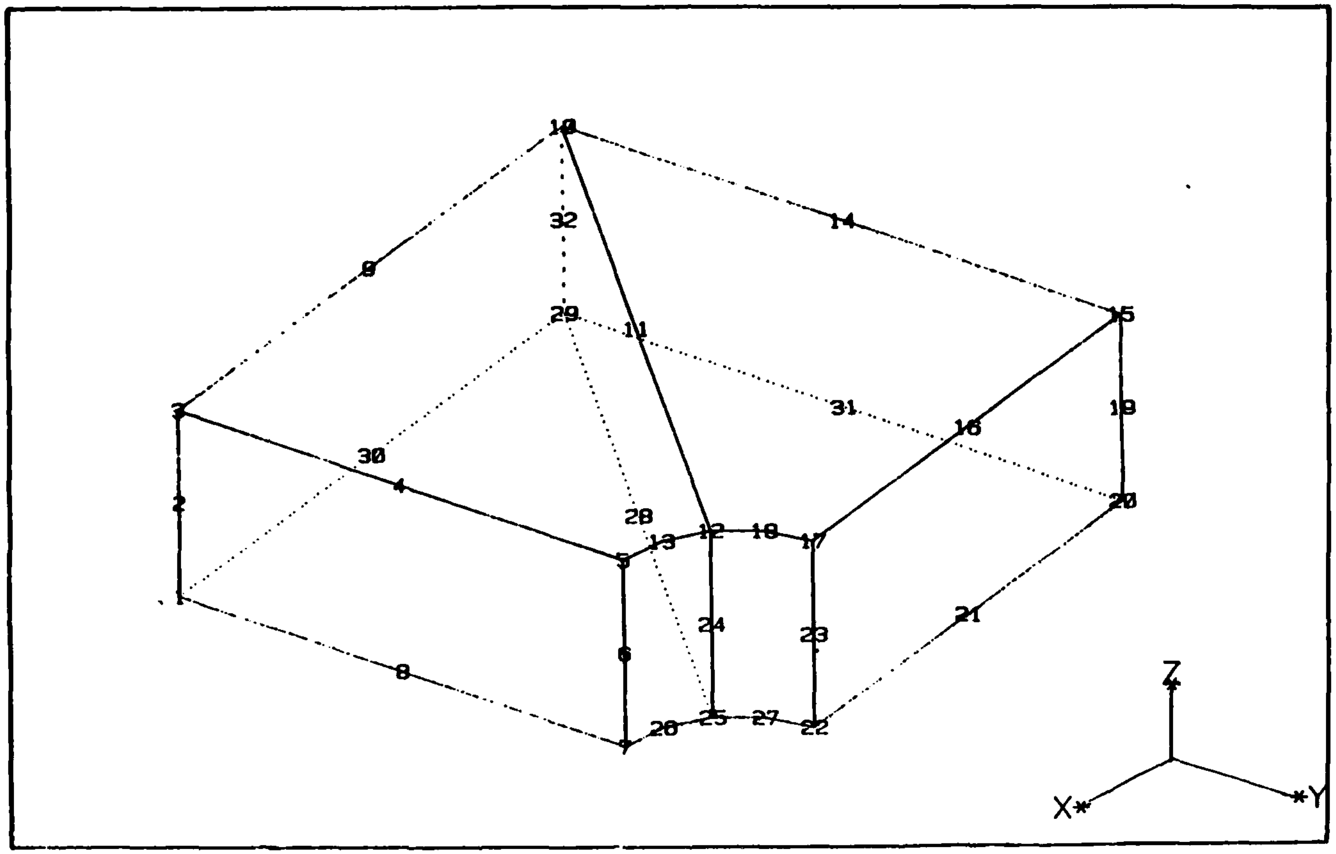


Fig 8 Preliminary mesh design and nodal numbering system of a quarter of the flat plate

Fig 9

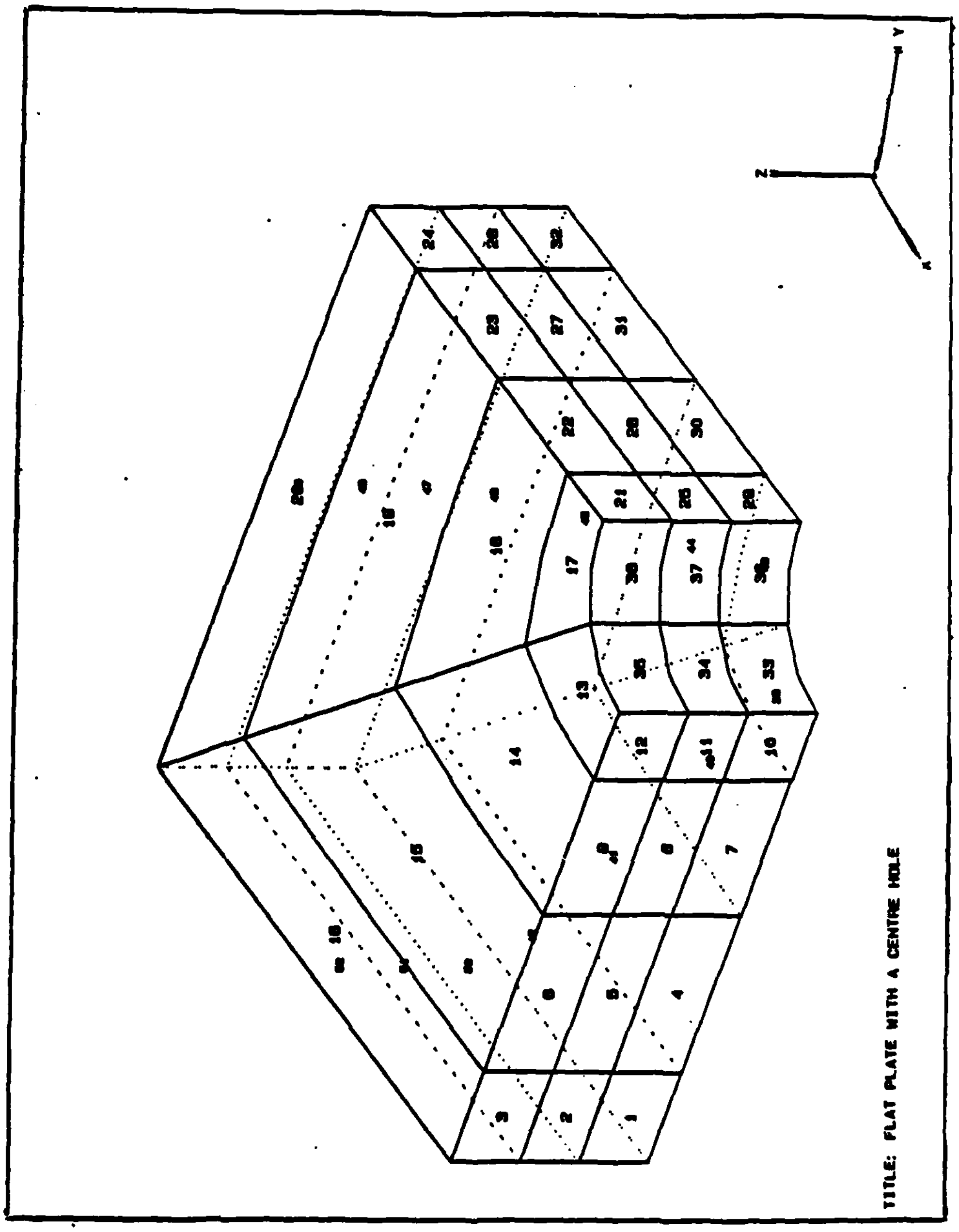


Fig 9 Mesh design and element numbering system of a quarter of the flat plate

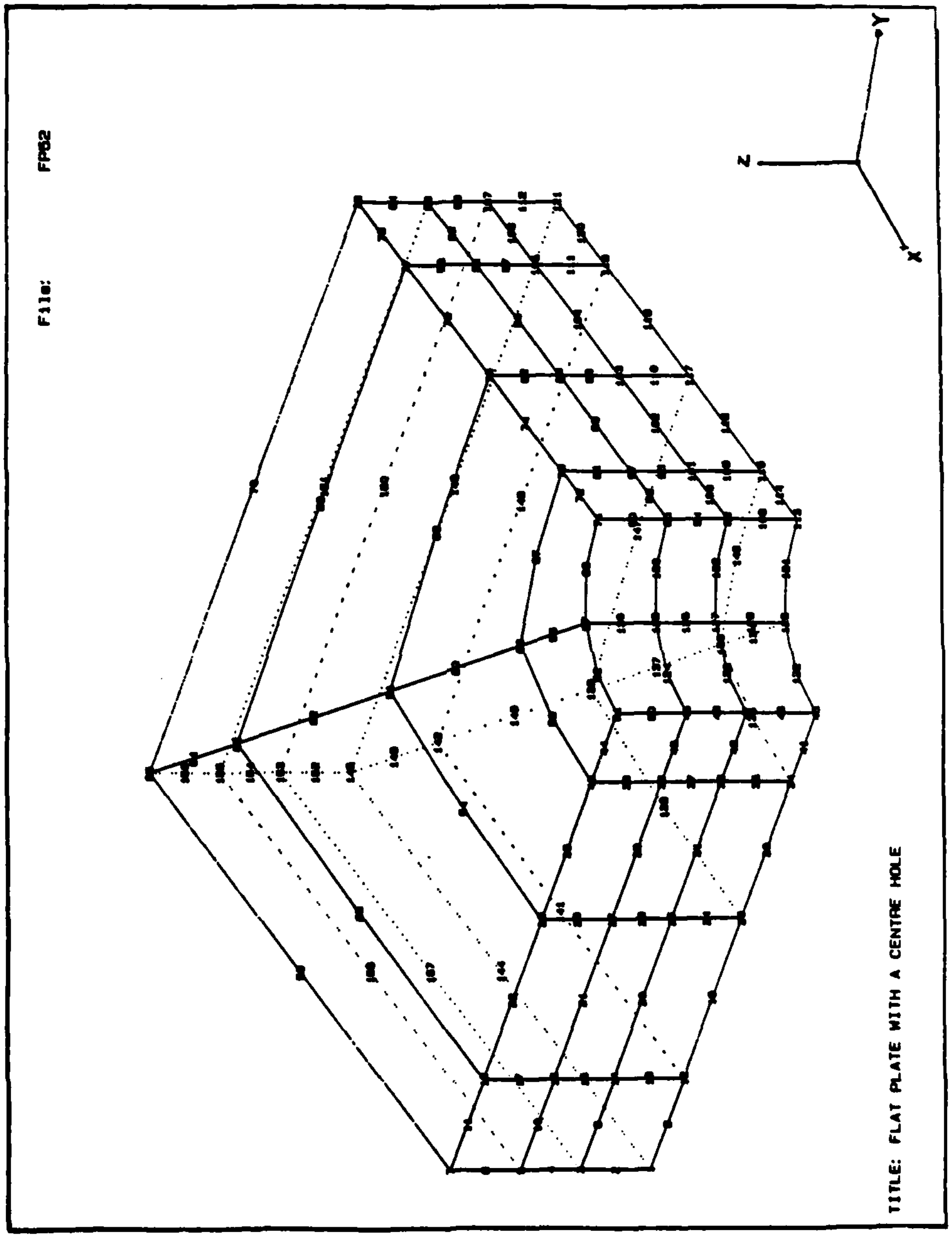


Fig 10 Mesh design and nodal numbering system of a ouarter of the flat plate

END

DATE
FILMED

2 82